Network Working Group
Internet-Draft
Expires: May 11, 2014

M. Andrews
ISC

November 7, 2013

Updating Parent Zones
draft-andrews-dnsop-update-parent-zones-04

Abstract

DNS UPDATE was developed to allow DNS zones to be updated.

There is a perception that UPDATE cannot be used in conjuction with the Registry, Registar, Registrant (RRR) model to update a zone.

This document explains how UPDATE can be used in the RRR model.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on May 11, 2014.

Table of Contents

1. Introduction 3
2. Requirements 3
3. Translation 3
4. Authentication 4
5. Direct to Registrar 4
6. Indirect to Registrar 4
7. UPDATE Server Discovery 5
8. Security Considerations 6
9. Normative References 6
Author's Address 6

1. Introduction

UPDATE [RFC2136] is designed to update any zone in the DNS. This includes updating delegating NS records, glue address records and DS records.

While UPDATE is primarily designed to UPDATE a zone directly there in no reason why UPDATE requests cannot be translated to the EPP requests to perform the changes.

This would provide a uniform model to update parent zone regardless of where they are in the DNS hierarchy or whether the zone is signed or not.

2. Requirements

This document was written with the following requirements in mind:

- must be able to authenticate the transaction.
- must be able to update address records to support automated renumbering.
- must be able to update DS records to support DNSKEY rollover buy key management tools.
- must work for unsigned zones (parent and/or child).
- must work for signed zones (parent and/or child).
- must work for RRR managed zones.
- must work for non RRR managed zones.
- desirable support updating of NS RRsets so that nameservers can ensure delegations delgation data remains consistent.

3. Translation

The Registrar would host a server that authenticates UPDATE requests received directly or relayed by the Registry using TSIG [RFC2845], then translate the actions in the UPDATE request into EPP transaction requests. The results of those EPP transactions would be relayed to the UPDATE client.

Requests that are not TSIG signed or fail verification must be rejected.

The translating server would handle a restricted subset of UPDATE requests, possibly ignoring the prerequiste section. UPDATE requests would be limited to those supported by EPP.

e.g. Add NS record. Delete all NS records. Add A record. Delete

AAAA record. Add DS record. Delete DS record.

The translating server may also override/ignore the TTL in the UPDATE request.

4. Authentication

Authentication would be done using TSIG. TSIG was designed to be used in a environment where requests are relayed.

Authentication can be done down to the <NAME,TYPE> tuple. There exist nameservers that already implement access contols down to this level of granularity based on the presented TSIG.

This would allow nameservers to update their own address records as they get renumbered without being able to update anything else.

This would allow DNSSEC key management software to update DS records without being able to update anything else.

As Registrars do all the authentication and generate the signed responses there is no need for the Registry to have access to the private key material used in TSIG.

Registrars already handle shared keys in these numbers with their web interfaces so it is not unreasonable to expect them to be able to handle a similar number of shared TSIG keys.

5. Direct to Registrar

The hardest part of Direct to Registrar is finding where to send the UPDATE request. This would most probably just be advised to the Registrant.

6. Indirect to Registrar

In the indirect model the Registry would host a UPDATE relay server which would examine the first record of the UPDATE section and relay the request to the Registrar of record for the owner name of that record. The Registrar would verify the validity if the request based on the TSIG then update the registry contents using EPP if appropriate. The response from the Registrar would be relayed back to the client via the Registry.

The Registry takes no action other than to relay the request and

response unless it is directed to do so by the Registrar.

The relay can use either TCP or UDP when forwarding UPDATE requests as TSIG supports changes to the DNS id field when a request/response is relayed. Only the Registrar and the client (Registrant) need to know the TSIG secret.

This is consistent with how tools like nsupdate work out where to send a UPDATE request if the zone is not explicity set. They look at the ownername of the first record and use it to discover the containing zone.

7. UPDATE Server Discovery

UPDATE server discovery is a issue when the RRR model is in use as the UPDATE may need to be directed through EPP and/or sent to a Registrar. There are a number of way this could be done:

1) Adding a underscore infix labels to the zone which contain SRV records at pointing to Registar/Registry servers for each child.

e.g. <child>._update._tcp.<parent> SRV 0 0 53 server.example.tld

The server pointed to could be be a relay server, as described above, or a UDPATE to EPP translating server. A relay server would allow for slower zone growth.

Using underscore infix labels requires no changes to nameservers operated by Registries but does require the zone content to be updated or a separate zone (e.g. _update._tcp.<parent>) to be delegated to contain this information.

A level of indirection could be added by using CNAME records to point to a domain operated by the registrar which contains the SRV record. This would allow the registrar to update the SRV records without having to update the zone being served by the registry. The CNAME would be updated on registrar changes. Note the target name the CNAME could also be managed by the registry as a way to consolidate the SRV record management.

```
child._update._tcp.tld CNAME registrar._registrars.tld
registrar._registrars.tld SRV 0 0 53 server.example.tld
```

As with traditional use of SRV, non-support can be signaled with

```
*._update._tcp SRV 0 0 0 .
```

If the Resistry is operating a relay this can be supported with a single wildcard record.

```
*._update._tcp SRV 0 0 0 server.registry.tld
```

The client can fallback to direct update to parent servers if no SRV record is discovered. This allows the scheme to work outside of the registry, registrar, registant model.

2) Extend UDPATE to return the update server. Currently the Zone section of the UPDATE refers to the zone to be update and is identified by the <QNAME,SOA,QCLASS> tuple. Replacing SOA with one or more of DS, NS, A and AAAA would allow a nameserver to distingish between a traditional UPDATE request and a request to find the UPDATE servers. The tuple would contain the resource to be updated and the reply would contain SRV records pointing to the UPDATE servers. As there would possibly more than one parent the owner records would refer to the parent zone being updated.

8. Security Considerations

The UPDATE requests are all TSIG signed. This is a proven method for securing UPDATE requests in the DNS.

9. Normative References

[RFC2136] Vixie, P., Thomson, S., Rekhter, Y., and J. Bound, "Dynamic Updates in the Domain Name System (DNS UPDATE)", RFC 2136, April 1997.

[RFC2845] Vixie, P., Gudmundsson, O., Eastlake, D., and B. Wellington, "Secret Key Transaction Authentication for DNS (TSIG)", RFC 2845, May 2000.

Author's Address

M. Andrews
Internet Systems Consortium
950 Charter Street
Redwood City, CA 94063
US

Email: marka@isc.org